µRFID

User Manual v1.00

Enable Your Design

ThaiEasyElec.com

On-Line Electronics Shop for Embedded System

Copyright © 2008, ThaiEasyElec , All rights reserved

Disclaimer:
This document is intended only to assist the reader in the use of the product. ThaiEasyElec.com shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

Index

1. Introduction **3**
2. Features **3**
3. Requirement 4
4. Drawing 5
5. Getting start **6**
5.1 Peripherals **6**
5.2 Power it on **6**
5.3 Creating project with Keil **9**
5.4 Source files description **11**
5.5 Operation **13**
5.6 Programming and Command line interface **14**

Appendix
1. Quick reference guide for port connections **17**

1. Introduction

uRFID is a RFID development board comes with NXP's powerful ARM7 LPC2103, 125KHz RFID reader ID-12, USB to serial bridge IC CP2102, 8kB EEPROM and serial number IC DS2411P. With the example software (open source code), the module can be configured as 2 modes: forward mode and access mode. With forward mode, the module acts as a simple USB RFID reader. With access mode, the module acts as a RFID access controller, which users can register tags easily with a command line 'add'. Up to 255 tags can be added since one tag use 32 bytes information and the EEPROM size is 8192 bytes (The last 32 bytes are reserved for configuration).

2 of 8 ports GPIO are by software, used as 'Hit' ports in access mode, one with active high and one with active low characteristic. These dedicated ports can be used to control a solenoid or relay for door opening purpose.

The on-board serial number IC DS2411P is a perfect embedded hard lock solution. Cooperating with PC application, user can lock there software to operate only with there authorized RFID reader.

2. Features

Hardware

- 125KHz RFID reader ID-12
- NXP's ARM7 LPC2103
- On board 8kB EEPROM (the last 32 bytes are reserved for configuration parameters)
- Serial number IC DS2411P
- 8 ports GPIO and 2 I2C pins
- USB to serial bridge IC CP2102 the MCU can be programmed with this serial port
- Mini-B USB connector (the board is powered from USB connector)

Software

Application software (the mode selected via console)
- Access mode (mode 0): a simple RFID access controller
- Forward mode (mode 1): a simple RFID reader with USB interface

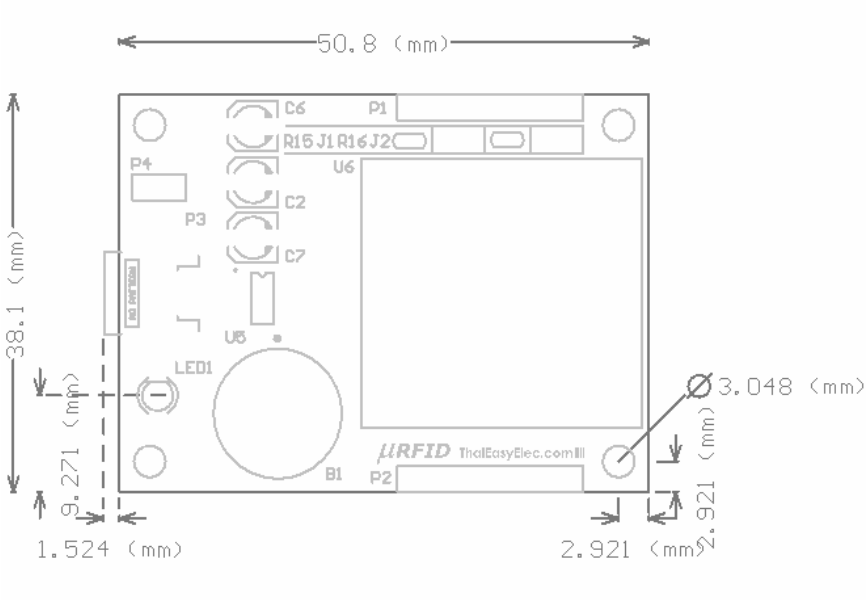
Low level drivers
- Serial port
- One wire interface
- I2C interface

3. Requirement

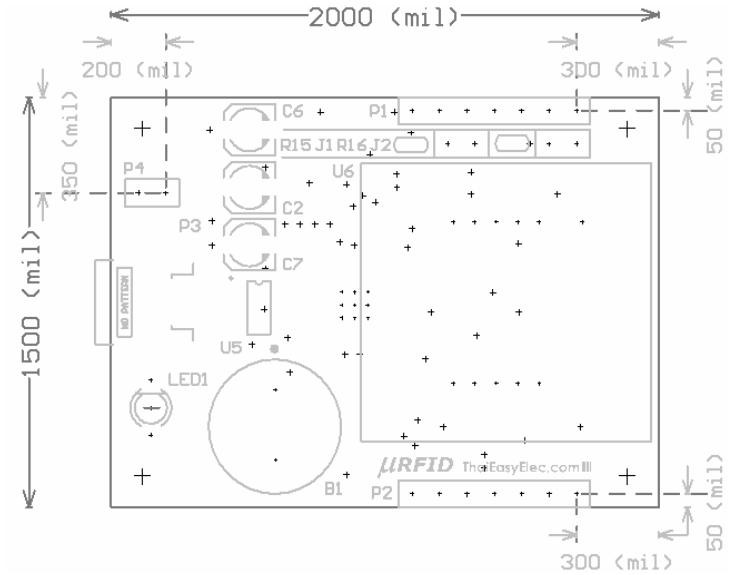
- USB cable (type A to mini-B) for supplying the board and programming
- 125KHz RFID tags
- CP2102 driver, can be downloaded from www.silabs.com/products/interface/usbtouart/Pages/default.aspx

4. Drawing

4.1 Outline drawing

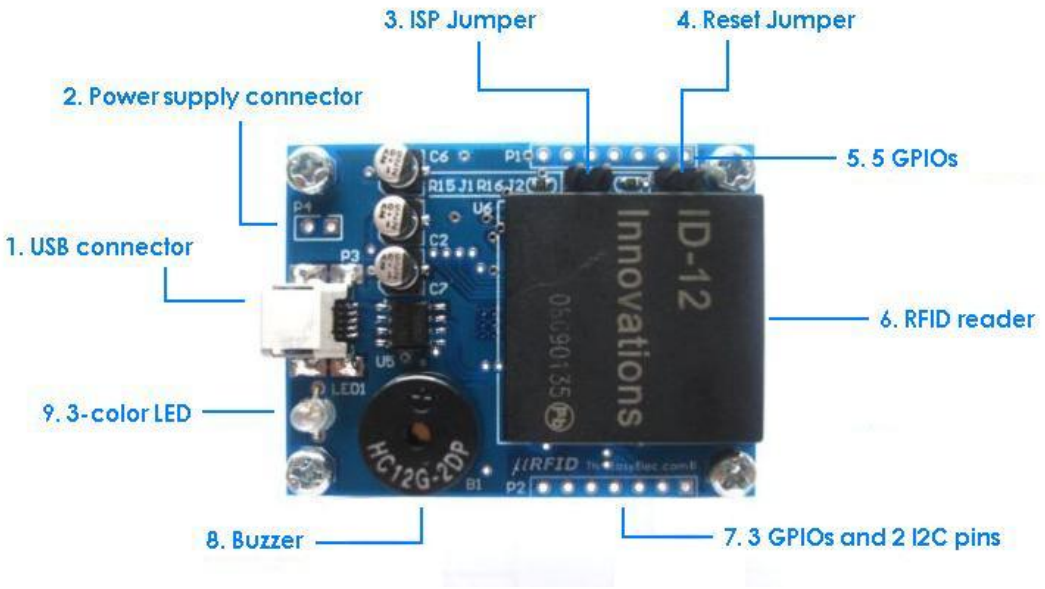


4.2 Drawing for docking purpose



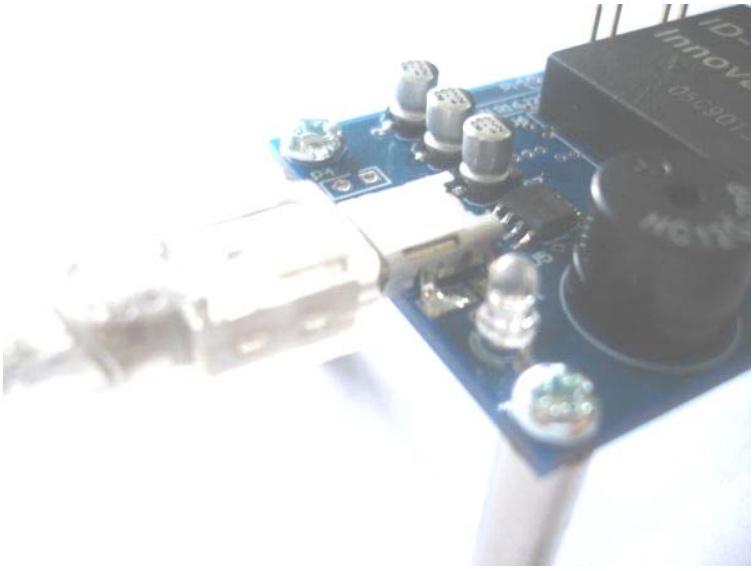
5. Getting start

5.1 Peripherals



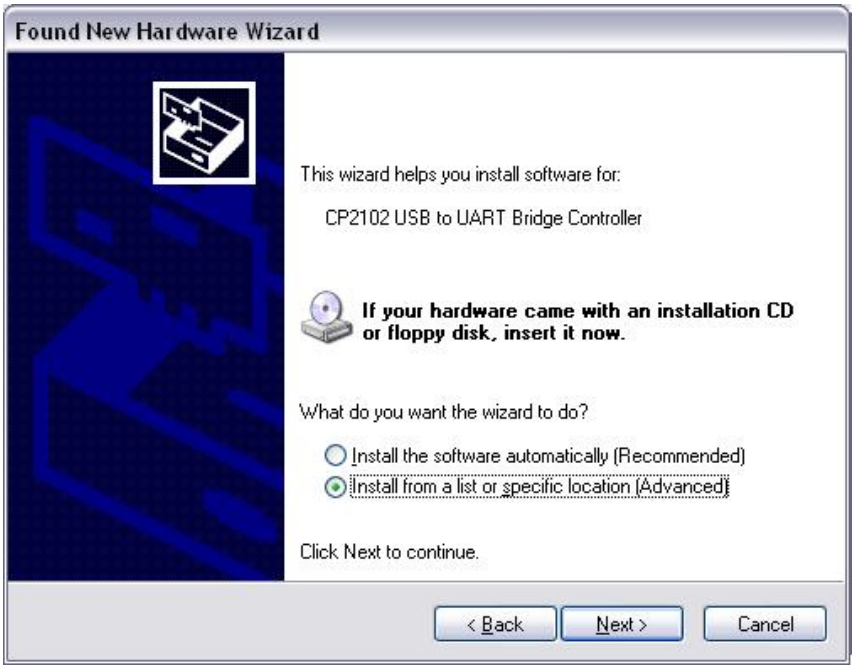
5.2 Power it on

Connect the board with your computer via USB cable as seen in picture.

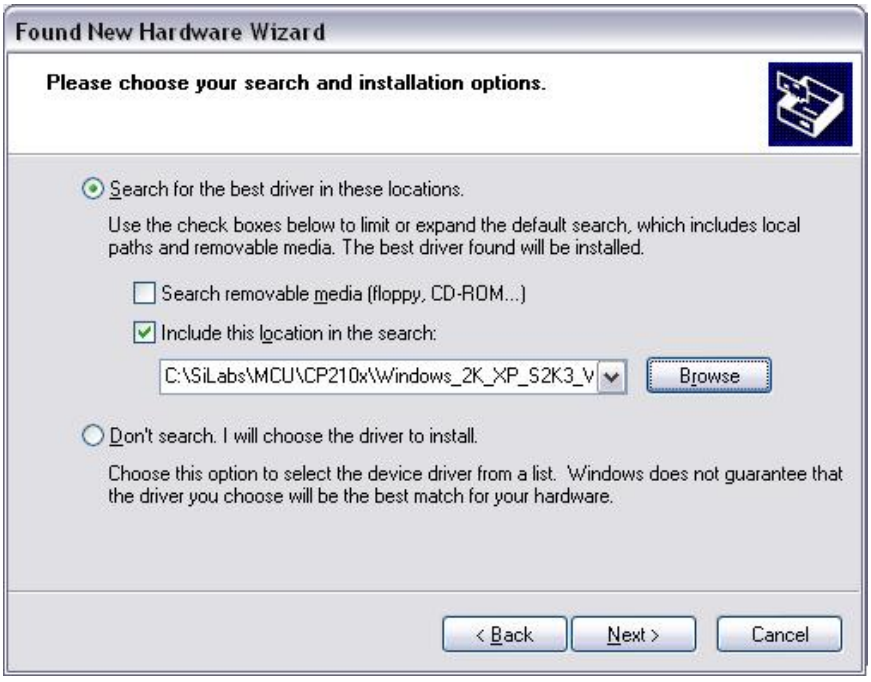


A ‘Found new hardware’ pop up should appear for the first time. Unless you install the driver downloaded from internet, please install it first. When the package is installed, follow the ‘Found new hardware wizard’.

Found New Hardware Wizard
Welcome to the Found New Hardware Wizard
Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).
Read our privacy policy
Can Windows connect to Windows Update to search for software?
Yes, this time only
Yes, now and every time I connect a device
No, not this time
Click Next to continue.
< Back
Next >
Cancel


Found New Hardware Wizard
This wizard helps you install software for:
CP2102 USB to UART Bridge Controller
If your hardware came with an installation CD or floppy disk, insert it now.
What do you want the wizard to do?
Install the software automatically (Recommended)
Install from a list or specific location (Advanced)
Click Next to continue.
< Back
Next >
Cancel

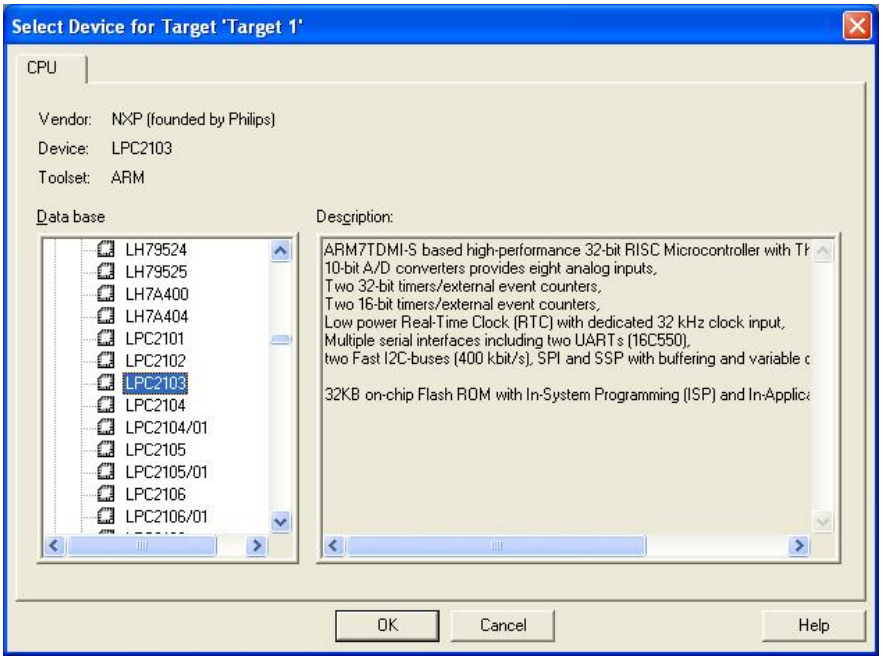
The path that contains driver is, by default: C:\SiLabs\MCU\CP210x\Windows_2K_XP_S2K3_Vista.


Found New Hardware Wizard
Please choose your search and installation options.
Search for the best driver in these locations.
Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
Search removable media (floppy, CD-ROM...)
Include this location in the search:
C:\SiLabs\MCU\CP210x\Windows_2K_XP_S2K3_V
Browse
Don't search. I will choose the driver to install.
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.
< Back
Next >
Cancel

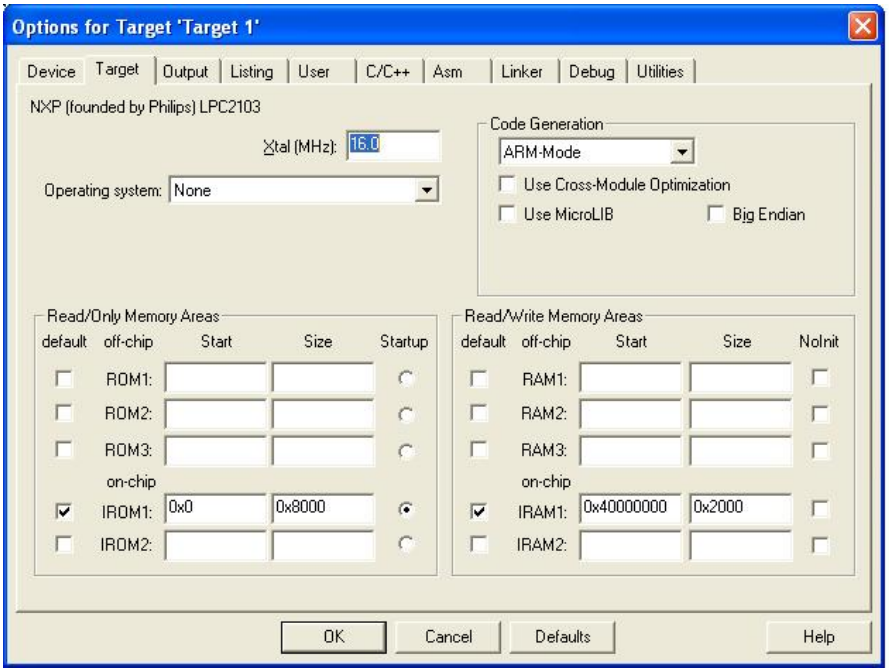
If the board starts up without ISP jumper installed, the LED should flash in orange color for 1 time per second.

5.3 Creating project with Keil

The board is supplied with codes. Users have to make project themselves. This guide is an example for Kei's software. Firstly, you have to create a project. And select the device as "LPC2103"

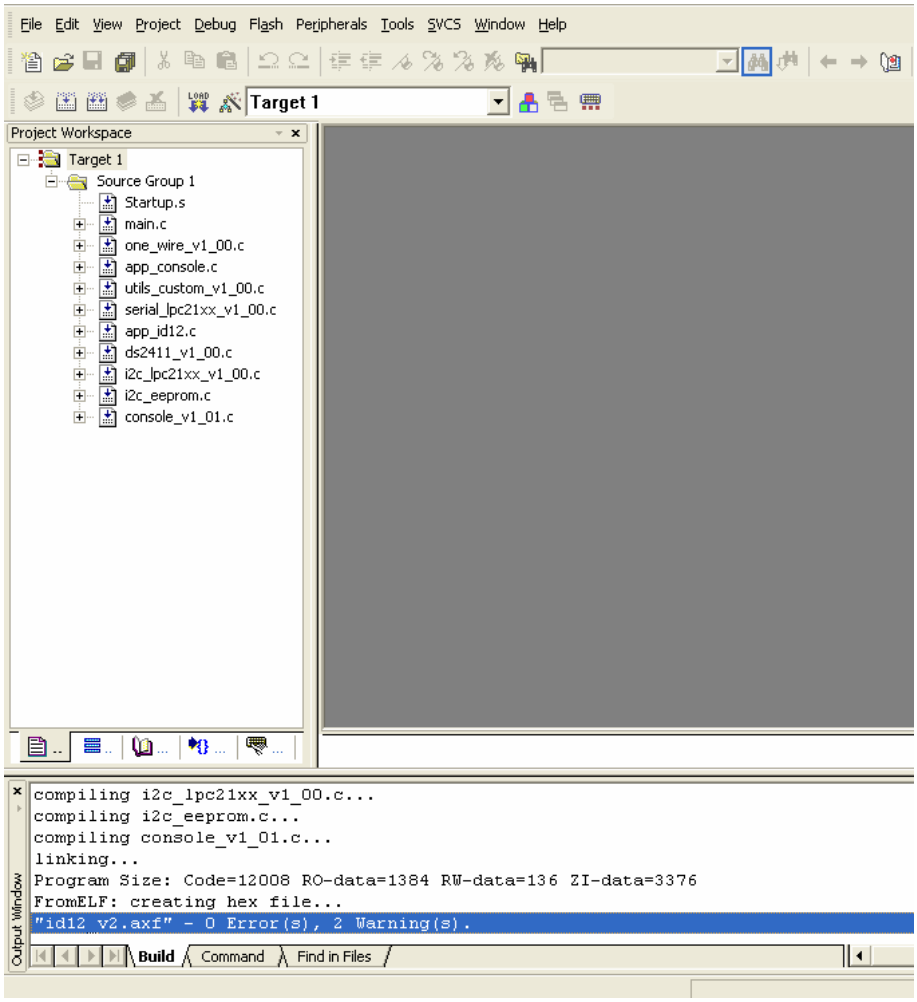


Secondly, set the XTAL frequency as 16 MHz.



Thirdly, replace the Startup.s file with the one you got from our website. And then add all other source files (.c) in the project.

After these settings, the project should be compiled without any errors (some warnings may occur). The project workspace should be seen as below.



5.4 Source files description

Header files (only important files are shown)

1. hw_urfid.h : define hardware, for example : LED1 is at port 0.4

```
#define LED1_DPRT        FIODIR
#define LED1_PRTS        FIOSET
#define LED1_PRTC        FIOCLR
#define LED1_PINP        FIOPIN
#define LED1_PIN         4
```

DPRT : direction port
PRTS : port set
PRTC : port clear
PINP : port input
PIN : pin number in the port

These may look awkward, but it's very easy to change some pins.

2. app_config.h : this file is used to configure many parameters various on each project. Header file including are in this file. And this file is included in most of source file. So which declaration shared on more than one source file should be declared on this file. This file also has function declarations from app_id12.c

3. All other header files are subjected to declare functions only.

Source files (only important files are shown)

1. main.c : background functions are here including timer interrupt service routine (T1_IRQHandler()), i/o initialization (io_init()). In case that user need to use more peripherals, PINSEL may be set in io_init(). Notice that AppInit() and AppRun() are called from main(), user application may be modified (or recreate) using these functions.

2. app_id12.c : the application software are here.

3. All other source files are hardware drivers, for example I2C, one-wire interface.

5.5 Operation

The example source code designed for 2 purposes, one as an access controller called access mode and two as a simple USB-interface RFID reader called forward mode.

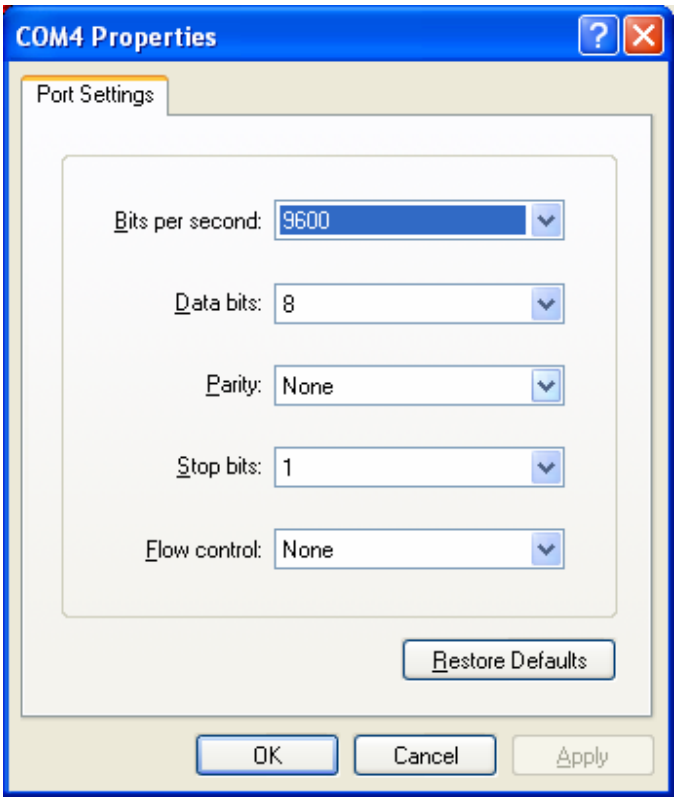
5.5.1 Access mode (mode 0)
With capability of up to 255 tags in a system, it suites for access control application or any others which one reader is enough. The module does checking sum, checking whether a tag is registered in the system. In addition, when a registered tag detected, the module generate 2 signals called ‘Hit’ ports. These ports are dedicated on P0.23 (active high) and P0.24 (active low) and a port is inverted to each other. In the other hand, when a tag is detected from reader and the tag’s serial number is in the list, the module generates a high pulse on P0.23 and in the same time, generates a low pulse on P0.24. The interval of these pulses can be configured using command ‘HIT_LONG’. User can configure this interval between 100 ms to 50 minutes.

5.5.2 Forward mode (mode 1)
In this mode, MCU does nothing with serial data receiving from ID-12 but forwards them to PC via USB Bridge. Suits for application in which, many of tags and readers are installed in the system. Only with this mode, an automatic generating buzz sound when tags detected can be disabled and let PC decides type of buzzing to be generated. These can be done with ‘AUTO_BEEP’ and ‘BEEP’ command, which will be described in next section.

5.6 Programming and Command line interface

To program the module, install a jumper on ISP jumper (see 5.1) and then plug the module to PC. Now the MCU should starts with ISP boot loader and will not flash LED. Open Flash magic and program the module. After that, uninstall the ISP jumper and reset the module by installing a jumper on reset jumper for a while, then take it off.

With the same USB cable and same serial port, user can communicate with the module using command line interface through HyperTerminal. Set the COM port as follow.



Every time the module starts up, it greets you with lines of text. The last of them says “enter 'help' for help”. Below is what you can see with ‘HELP’ command. Remember that this command line interface is not case-sensitive.

AUTO_BEEP n (1=on,0=off) : turn on/off auto beep
BEEP n (1=once,2=twice) : beep manually
READ_SN : read module's serial number
MODE n (1=fwd,0=access) : select operation mode
ECHO n (1=on,0=off) : turn on/off console echo
ADD card_label (string) : add a card
LIST : list all added card
REM nn : remove card at nn
REM_ALL : remove all cards
HIT_LONG : set Hit interval (in 100ms)
DEFAULT : load default setting
SAVE : save setting
SHOW : show setting

'AUTO_BEEP' command sets whether you want the module to generate buzz sound when a tag detected, only applicable with Forward mode.

If you (or your PC application) wish to generate buzz sound, use 'BEEP' command. It, in the same time, flash LED in different color.

'READ_SN' command allows you to read serial number from DS2411P.

'MODE' command selects which mode to be operated.

'ECHO' command allows PC application not to process much data by turning echo off.

'ADD' command follow with tag label, for example, entering 'add tag1' then put a tag on RFID reader. The tag will be registered and this label (string "tag1") will be printed out to serial port whenever this tag is detected again.

'LIST' commands allow user to see the tag list.

'REM' and 'REM_ALL' command allow user to remove a tag or remove all tag in the list.

‘HIT_LONG’ command sets the interval time in 100 ms for ‘Hit’ event.

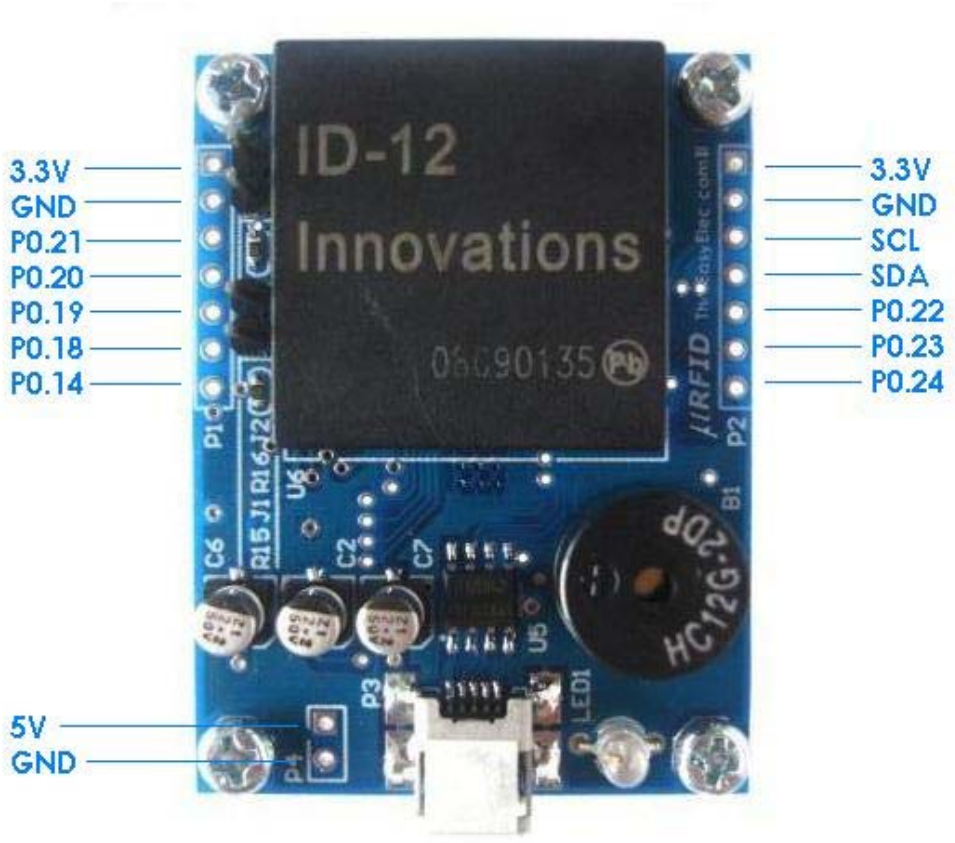
‘DEFAULT’ command loads default setting to the module which the module operates in Forward mode, echo on and automatic beep on.

‘SAVE’ command let user to save there setting to EEPROM including operating mode, echo and automatic beep behaviors. Doing things with the tag list is automatically saved to EEPROM.

‘SHOW’ command let user to see current setting on the module.

Appendix

1. Quick reference guide for port connections



P1 contains 7 pins, including 5 GPIOs and power supply pins.

Pin	MCU Pin	Available function(s)
1	3.3V	-
2	GND	-
3	P0.21	GPIO,SSEL1,MAT3.0
4	P0.20	GPIO,MAT1.3,MOSI1
5	P0.19	GPIO,MAT1.2,MISO1
6	P0.18	GPIO,CAP1.3
7	P0.14	GPIO,SCK1,EINT1

Also, P2 contains 3 GPIOs and 2 dedicated I2C pins.

Pin	MCU Pin	Available function(s)
1	3.3V	-
2	GND	-
3	P0.2	SCL
4	P0.3	SDA
5	P0.22	GPIO,AD0.0
6	P0.23	GPIO,AD0.1
7	P0.24	GPIO,AD0.2